

<https://doi.org/10.7236/IIBC.2017.17.3.137>

IIBC 2017-3-16

IoT 카스토퍼 기반 스마트 주차안내 시스템

Smart Parking Guidance System based on IoT Car-stoppers

심동하*, 양지훈**, 손정기***, 한승한**, 이현민**

Dongha Shim*, Ji-Hoon Yang**, Jeungki Son***, Seung-Han Han**, Hyounmin Lee**

요 약 IoT 카스토퍼 기반의 스마트주차 안내 시스템을 제안하고 구현하였다. IoT 센서 모듈을 내장한 카스토퍼 감지센서는 기존의 매립형 주차센서에 비해 시공이 용이한 장점을 가진다. 순차적 점대점 통신을 통해 카스토퍼에서 전송된 주차상태 데이터는 IoT 게이트웨이를 통해 웹서버에 저장되고, 스마트기기의 안드로이드 앱을 통해 원격으로 주차공간을 확인할 수 있다. 배터리로 구동되는 IoT 카스토퍼는 소모 전력을 줄이기 위해 위치독 타이머와 연동한 액티브/슬립 사이클 방식을 사용한다. 카스토퍼의 소모전력은 액티브 모드와 슬립 모드에서 각각 80mW와 25mW로 측정되었다. 슬립 모드에서 발생하는 전력 소모를 최소화 할 수 있는 초저전력 IoT 센서모듈 구조를 제안하였다. 실제 주차장에서 구현된 시스템의 동작을 검증하였다.

Abstract This paper presents a smart parking guidance system based on IoT car-stoppers. The car-stopper embedding an IoT sensor module has the advantage of easy installation compared to conventional parking sensors buried in ground. The parking status data are transferred to the IoT gateway by the sequential point-to-point communication between the car-stoppers. The data transferred from the IoT gateway are stored in the web server, and parking spaces can be monitored remotely through the Android app in a smart device. An active/sleep cycle method using a watch dog timer is employed to reduce the power consumption of the battery powered car-stopper. The power consumption of the car-stopper is measured to be 80 and 25 mW at the active and sleep mode, respectively. A configuration of ultra-low-power IoT sensor module is proposed to minimize the power consumption in the sleep mode. The operation of the implemented system has been verified in a real-world parking lot.

Key Words : IoT, Car-stopper, Smart parking, Active/sleep cycle, Low-power

I. 서 론

최근 전 세계 대도시를 중심으로 차량 증가에 따른 주차공간 부족 문제가 점점 심각해지고 있다. 운전자들은 주차 공간을 찾기 위해 시간과 연료를 낭비하게 되고 나아가 교통체증이나 사고를 유발하기도 한다. 이러한 문제를 해결하기 위해 최근 바르셀로나(스페인), 미국 샌프

란시스코, 런던(영국), 두바이(UAE), 멜버른(호주) 등지에서 IoT를 접목한 스마트 주차 안내 시스템을 운영하고 있다^[1,2]. 또한 다양한 국내외 회사들이 IoT 기반 주차 시스템을 선보이고 있다. 대표적인 해외회사로 Libelium, Nwave, PNI Sensor, Smart Parking 등이 있으며 국내회사는 ParkingCloud, ino-on, JND, MARGO 등이 있다^[3-10].

*정회원, 서울과학기술대학교 MSDE전공

**준회원, 서울과학기술대학교 스마트생산융합시스템공학과

***준회원, 서울과학기술대학교 MSDE전공

접수일자: 2017년 5월 3일, 수정완료: 2017년 6월 3일

게재확정일자: 2017년 6월 9일

Received: 3 May, 2017 / Revised: 3 June, 2017 /

Accepted: 9 June, 2017

*Corresponding Author: dongha@seoultech.ac.kr

MSDE Programme, SeoulTech, Korea

현재 상용화된 시스템들은 대부분 바닥을 파서 감지 센서를 매립하기 때문에 시공이 복잡한 단점을 가지고 있다. 이러한 문제점을 개선하기 위해 본 논문에서는 IoT 센서모듈이 내장된 카스토퍼(car-stopper) 기반의 시공이 용이한 주차안내 시스템을 제안하였다. 그림 1은 제안된 카스토퍼 기반의 스마트 주차안내 시스템의 개요이다. 시스템은 크게 IoT 카스토퍼, IoT 게이트웨이(gateway), 웹서버(web server), 안드로이드앱(Android app)으로 구성된다. 카스토퍼에 내장된 IoT 센서모듈에서 얻어진 주차 데이터는 순차적 점대점(sequential point-to-point) 통신을 통해 카스토퍼에서 카스토퍼로 전달된다. 취합된 주차 데이터는 IoT 게이트웨이를 거쳐 웹서버로 저장되며, 스마트기기를 이용해 웹서버에 접속하여 안드로이드앱을 통해 원거리에서 주차공간의 유무를 파악할 수 있다. 배터리의 동작시간을 늘리기 위해 IoT 카스토퍼는 주기적으로 센싱 및 통신을 수행하는 액티브/슬립 사이클(active/sleep cycle) 방식을 사용한다 [11].

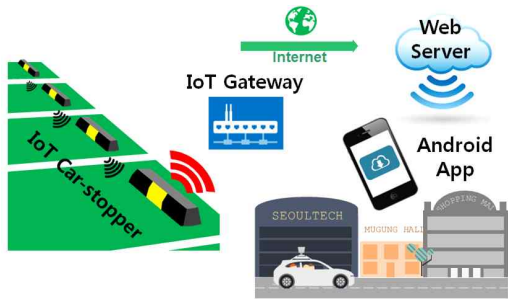


그림 1. 제안된 IoT 카스토퍼 기반의 스마트 주차안내 시스템 개요
Fig. 1. Overview of proposed smart parking guidance system based on IoT car-stoppers

II. 시스템 설계

그림 2은 스마트 주차 안내 시스템의 아키텍처를 보여준다. 카스토퍼 안에 설치된 IoT 센서모듈은 차량을 감지해 정보를 Zigbee 모듈을 통해 무선으로 이웃한 카스토퍼로 전달한다. 카스토퍼를 거치면서 수집된 주차 데이터들은 IoT 게이트웨이(gateway)로 전송되고 Wi-Fi 및 인터넷을 통하여 웹서버로 전송된다. 사용자는 안드로이드앱(Android app)을 이용해 웹서버(web server)에 접속하여 주차공간을 확인할 수 있다. Zigbee 네트워크는 기

본적으로 엔드디바이스(end device), 라우터(router), 코디네이터(coordinator)로 구성되며 카스토퍼는 엔드디바이스에 해당한다[12]. 엔드디바이스를 통해 수집된 주차 데이터는 라우터와 코디네이터를 통해 IoT 게이트웨이로 전송된다.

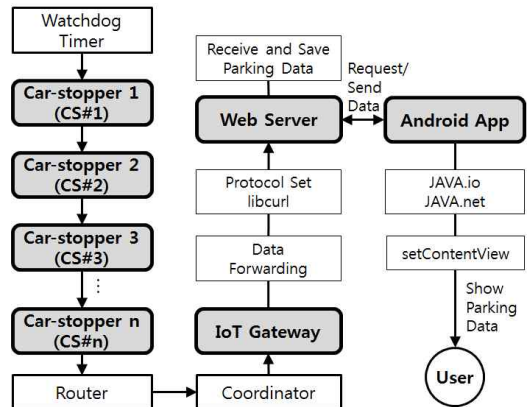


그림 2. 제안된 스마트 주차안내 시스템 아키텍처
Fig. 2. Architecture of the proposed smart parking guidance

카스토퍼에 내장된 IoT 센서모듈은 그림 3과 같이 라디오(radio), MCU(microcontroller unit), 센서(sensor), 배터리(battery)로 구성되어 있다. 차량의 유무에 따라 변하는 IR센서의 출력신호를 MCU가 감지해 이웃한 카스토퍼의 IoT 센서모듈로 주차 데이터를 전송한다. 신호는 소모전력을 줄이기 위해 작은 MCU와 라디오로 ATmega8과 ZigBee 방식의 XBee 모듈을 각각 사용하였다[13,14]. 센서는 배터리에 직접 연결되어 있지 않고 MCU로부터 전원을 공급받는 구조로 MCU가 활성화된 동안만 전력을 소모하게 된다.

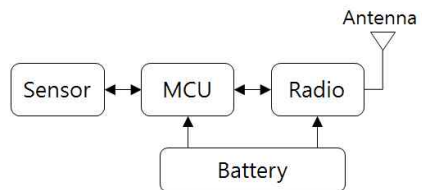


그림 3. IoT 센서모듈 블록 다이어그램
Fig. 3. Block diagram of IoT sensor module

ZigBee는 무선네트워크를 위해 개발된 통신 프로토콜

이다. 다른 통신방식에 비해 가격이 싸고 확장성이 높으며 전력을 적게 소모하는 장점을 가진다. 배터리로 구동되어 저전력 특성이 요구되는 카스토퍼에 적합하다. 일반적으로 ZigBee 네트워크는 트리 토폴로지(tree topology)나 메시 토폴로지(mesh topology)가 많이 사용된다^[12]. 엔드디바이스는 라우터에 연결되며 라우터를 통해 통신이 이루어진다. 주차장과 같이 엔드디바이스가 넓은 영역에 분포할 경우 엔드디바이스와 라우터 사이의 통신거리 제한으로 다수의 라우터가 요구 된다.

제안된 시스템은 라우터의 개수를 줄이기 위해 점대점(point-to-point) 통신방식을 확장한 엔드디바이스에서 이웃한 엔드디바이스로 직접 데이터를 전송하는 순차적 점대점(sequential point-to-point) 통신 방식으로 구현되었다(그림 4). 특히 주차장의 경우 카스토퍼가 가까이 배치된 경우가 많아 엔드디바이스 사이의 원활한 통신이 가능하다. 이러한 비동기(asynchronous) 전송 방식은 속도가 느리고 지연이 발생하는 단점이 있지만 빠른 업데이트가 요구되지 않는 주차 시스템에서는 문제가 되지 않는다.

그림 4는 주차 데이터의 전송과정을 보여준다. 주차공간이 비어 있을 경우 주차비트(parking bit) 0(vacant)을, 차량을 감지한 경우 주차비트 1(occupied)을 발생한다. 이전 카스토퍼에서 전송된 주차 데이터 배열에 현재 카스토퍼의 주차비트가 추가(concatenation) 되어 다음 카스토퍼로 전달한다. 카스토퍼를 거치며 축적된 주차데이터 배열은 최종 카스토퍼에 내장된 라우터를 통해 IoT 게이트웨이에 연결된 코디네이터로 전달된다.

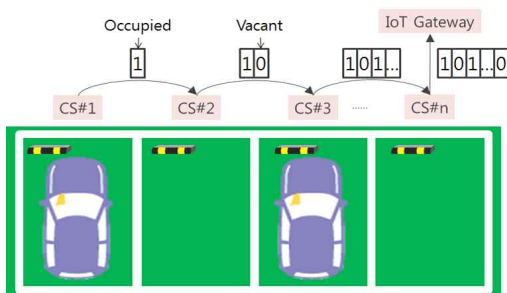


그림 4. 스마트 주차안내 시스템의 주차데이터 전송
 Fig. 4. Parking data transmission in the smart parking guidance system

일반적으로 주차 상황은 시간 간격을 두고 변하는 경우가 많기 때문에 센싱과 통신을 이루어지는 시간만 카

스토퍼를 액티브 모드로 유지 하는 슬립/액티브 싸이클 방식을 이용하면 소비전력을 줄일 수 있다^[11]. 그림 5는 저전력 동작을 위해 카스토퍼에 적용된 슬립/웨이크 스케줄링(scheduling)을 보여준다. 첫 번째 카스토퍼(CS#1)의 MCU는 일정한 주기(t_{ip})로 펄스를 발생하는 워치독 타이머에 의해 활성화 되고 센서를 통해 주차상태를 감지한다. 이어서 CS#2에 웨이크 신호를 보내 MCU를 활성화 시키고 주차데이터를 전송한 후 슬립신호를 받고 슬립모드로 들어간다. 이 과정을 반복하며 주차데이터가 순차적으로 이웃한 카스토퍼로 전달된다. 라우터와 연결된 마지막 카스토퍼(CS#n)의 MCU는 라우터에 주차데이터 배열을 전송하고 슬립 신호를 수신해 슬립 모드로 들어간다. MCU와 MCU에서 전원을 공급받는 센서는 슬립 모드에서 전력을 거의 소모하지 않는다. 반면 카스토퍼의 라디오는 항상 액티브 모드를 유지한다.

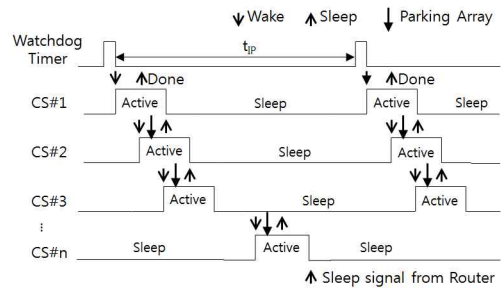


그림 5. 저전력 동작을 위한 슬립/웨이크 스케줄링
 Fig. 5. Sleep/Wake scheduling for low-power operation

저전력 동작을 위한 카스토퍼의 동작 시퀀스는 그림 6과 같다. (1) 카스토퍼 k (CS# k)의 MCU는 카스토퍼 $k-1$ (CS# $k-1$)으로부터 웨이크업 신호를 받고 액티브 모드가 된다. (2) CS# $k-1$ 로부터 주차데이터 배열을 전송받는다. (3) 슬립신호를 보내 CS# $k-1$ 을 슬립 모드로 만든다. (4) 센서를 통해 주차여부를 검출하고 해당 주차비트를 주차데이터 배열의 오른쪽 끝에 추가(concatenation)한다. (5) 웨이크 신호를 보내 CS# $k+1$ 의 MCU를 액티브 모드로 만든다. (6) UART 통신을 이용해 카스토퍼 CS# $k+1$ 로 주차데이터 배열을 전송한다. (7) CS# $k+1$ 에서 슬립 신호를 전송받고 슬립 모드로 들어간다.

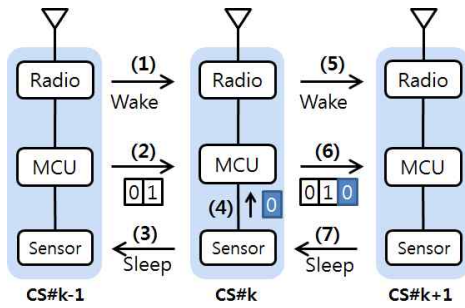


그림 6. IoT 카스토퍼 동작 시퀀스
Fig. 6. Operating sequence of IoT car-stoppers

III. 시스템 구현

1. IoT 카스토퍼

차량감지를 위한 센서는 IR 근접센서(GP2Y0A21YK, Sharp)를 사용하였다^[15]. 두개의 적외선 센서를 사용하여 이중감지로 검출의 정확도를 높였다. 감지거리는 센서모듈에서 50cm로 세팅 되었다.

센서 신호 처리를 위해 저전력 MCU인 ATmega8을 사용하였다^[13]. ATmega8은 시스템의 구현에 필요한 프로그램 메모리, ADC, UART 통신 등의 기능을 제공한다. ATmega8은 5개 슬립모드를 가지고 있으며 이중 가장 적은 전력을 소모하는 파워다운(Power-down) 모드를 사용했다. UART 통신을 이용해 라디오 모듈과 통신한다. 라디오는 ZigBee 통신 기반의 저전력 XBee 모듈을 사용했으며, 설정 프로그램인 X-CTU를 이용해 XBee 모듈의 세팅을 진행하였다. XBee 모듈은 UART 통신으로 ATmega8과 송수신 한다. 배터리는 3.7V/2800mAh 용량의 리튬이온 충전지(ICR-18650, Samsung)를 사용하였다^[16]. DC-DC 컨버터와 레귤레이터(LM1117, Linear Technology)를 이용해 배터리의 출력 전압을 5V와 3.3V로 변환 후 각각 MCU와 XBee 모듈에 공급하였다^[17].

첫 번째 카스토퍼(CS#1)에 웨이크 신호를 전송하는 위치독 타이머는 Texas Instruments사의 초저전력 타이머인 TPL5010을 사용하였다^[18]. 펄스의 주기(t_{p})는 외부 저항 값을 조정해 1초에서 최대 2시간 사이의 값으로 세팅할 수 있다. 20kohm의 외부저항을 연결해 위치독 타이머의 주기(t_{p})를 50초로 세팅하였다. 2.5V/35nA를 전원으로 사용하는 TPL5010은 초저전력 동작이 요구되는 다양한 WSN(wireless sensor networks)이나 IoT 응용에

활용될 수 있다.

IoT 센서모듈의 각 부품을 고정하기 위하여 3D 프린터를 이용해 플라스틱 케이스를 제작하였다. 그림 7은 케이스에 고정된 IoT 센서모듈을 보여준다. 종이를 이용해 그림 8과 같이 카스토퍼 모형(730x140x115mm³)을 제작하고 내부에 IoT 센서모듈을 장착하였다. 윈도우(window) 안쪽에 방수를 위해 유리 커버를 설치하고 센서를 배치하였다. 실제로 윈도우의 유리 표면에 물이 묻어도 센서가 정상 작동하는 것을 확인할 수 있었다.

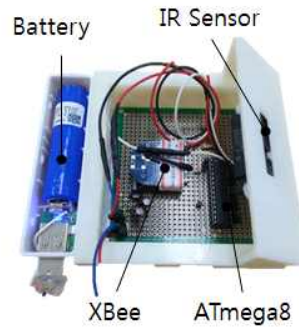


그림 7. IoT 센서모듈 사진
Fig. 7. Photo of IoT sensor module

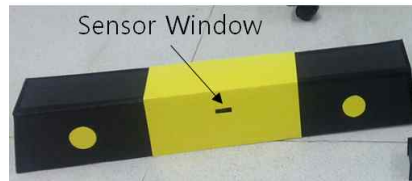


그림 8. IoT 카스토퍼 모형
Fig. 8. IoT car-stopper model

2. IoT 게이트웨이

IoT 게이트웨이는 카스토퍼로부터 주차상태 데이터를 수신하여 웹서버로 전송하는 역할을 한다. 초소형 싱글보드 컴퓨터인 라즈베리 파이(Raspberry Pi)를 이용해 IoT 게이트웨이를 구현하였다^[18]. 수신된 주차상태 데이터는 라즈베리파이에서 데이터 전송 라이브러리인 libcurl을 통해 웹서버로 전송된다^[19]. IoT 게이트웨이는 XBee 코디네이터를 포함해 카스토퍼 CS#n에 포함하며 라우터를 통해 주차 데이터를 전송 받는다.

3. 웹서버

퍼블릭 클라우드 서비스인 AWS(Amazon Web Services)에서 제공하는 EC2(Elastic Compute Cloud) 틀을 이용해 웹서버를 제작 하였다^[20]. AWS 기반의 웹서버를 이용하면 어디서나 IoT 주차 안내 시스템의 이용이 가능한 장점이 있다. 또한 여러개의 IoT 게이트웨이와 동시에 접속 할 수 있기 때문에 공간적으로 멀리 떨어진 주차장의 상태를 동시에 확인할 수 있다.

웹서버는 IoT 게이트웨이로부터 수신된 주차상태 데이터를 시간정보를 함께 저장한다. 사용자의 스마트기기에서 요청을 받으면 웹 서버는 Wi-Fi를 통해 저장된 데이터를 전달한다. 우분투(Ubuntu) 기반의 Apache2 웹서버를 사용했으며, 리눅스(Linux) 웹서버의 기본 언어인 PHP를 이용해 웹개발을 수행하였다^[21,22]. MYSQL를 이용해 주차상태 데이터를 DB(database)로 구성하고 PHP 프로그래밍을 통해 데이터를 추출해 웹페이지로 출력하였다^[23].

4. 안드로이드 앱

안드로이드 스튜디오(Android Studio)를 이용해 주차 안내 앱(App)을 제작 하였다^[24]. 스마트기기는 서버에서 전달된 데이터를 안드로이드 앱을 통해 주차상태 데이터를 디스플레이 해준다. XML을 이용해 클래스나 오브젝트를 위한 레이아웃을 형성하고 JAVA를 이용해 함수를 작성하였다. 안드로이드 개발환경은 Android 4 (API level 14)을 지원하며, JAVA.io와 JAVA.net 패키지를 이용해 서버와 소켓 통신을 구현 하였다. ActionBar 기능을 이용해 UI의 일관성을 높여 탐색과 뷰전환을 용이하게 하였다. 그림 9와 같이 차량의 유무에 따라 해당 구역의 색깔이 다르게 표시된다.

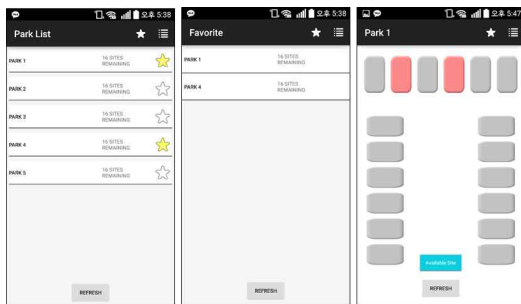


그림 9. 주차안내 안드로이드 앱 화면
 Fig. 9. Android app displays for the parking guidance app

IV. 실험 및 결과

제작한 스마트 주차시스템의 평가를 위해 건물 지하 주차장에서 실험을 진행 하였다(그림 10). IoT 게이트웨이는 주차장 기둥의 2.1m 높이 위치에 설치되었으며, 이때 전송거리는 주변환경과 장애물의 상황에 따라 20-30m의 범위로 측정되었다. 카스토퍼는 차량의 유무를 정상적으로 감지하였고, IoT 게이트웨이와 웹 서버를 통해 전송된 주차상태 데이터를 안드로이드 앱을 통해 확인할 수 있었다. 주차장 주변 반경 2km 이내의 다양한 위치에서 업데이트에 걸리는 지연시간을 측정한 결과 11초 이내로 측정 되었다.

5V 전원에는 대해 액티브 모드인 경우 카스토퍼의 전류는 16mA로 측정되었다. 소모 전력은 80mW이며 ATmega8, 센서, XBee 모듈이 각각 31%, 38%, 31%를 소모하는 것으로 예측된다. 슬립 모드의 경우 카스토퍼의 전류는 5mA으로 25mW의 전력을 소모한다. MCU와 센서는 슬립 모드에서 전력을 거의 소비하지 않기 때문에 XBee 모듈의 소모 전력이 전체 소모 전력이 된다. 슬립 모드에서 액티브 모드에 비해 소모 전력이 31%로 감소하였다.



그림 10. 설치된 카스토퍼 모듈(좌) 및 안드로이드 앱 화면(우)

Fig. 10. Installed car-stopper module (left) and Android App display (right)

구현된 시스템에서 MCU와 달리 XBee 모듈은 항상 액티브모드로 동작하며 전력을 소모한다. 그림 11의 초저전력 IoT 센서모듈 구조를 이용하면 이 문제를 해결할 수 있다^[25]. 라디오와 MCU가 위치독 타이머에 의해 제어되는 스위치를 통해 배터리와 연결된 구조로 개별 카스

토퍼가 타이머를 포함하게 된다. 모든 위치독 타이머 주기를 동일하게 하고 동기화(synchronization) 시키면 모든 카스토퍼의 액티브 및 슬립 모드를 동기화 시킬 수 있다. 액티브 모드에서 위치독 타이머가 스위치를 ON 시키면 카스토퍼들이 동시에 주차상태 데이터를 전송하고, 슬립 모드에서 스위치가 OFF 되면 라디오와 MCU의 전원이 동시에 차단되기 때문에 전력 소모가 발생하지 않는다. 개방상태의 스위치에 누설전류(leakage current)가 흐르면 전력 소모를 유발하기 때문에 저누설 스위치(low leakage switch)를 사용해야 한다.

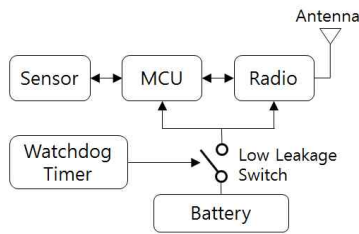


그림 11. 초저전력 IoT 카스토퍼 블록도
Fig. 11. Block diagram of Ultra-low-power IoT car-stopper module

V. 결 론

본 논문에서는 IoT 카스토퍼 기반의 시공이 용이하고 사용이 편리한 스마트 주차 안내 시스템을 제안하고 구현 하였다. 센서와 통신 모듈을 내장한 IoT 카스토퍼의 순차적 점대점 통신을 통해 취합된 주차상태 배열 데이터는 IoT 게이트웨이를 통해 웹서버로 전송된다. 주차장에서 실험을 통해 개발된 안드로이드 앱을 이용해 웹서버에 저장된 주차공간 정보를 확인할 수 있었다. 소모 전력 감소를 위해 MCU와 센서를 위치독 타이머를 이용해 액티브/슬립 싸이클 방식으로 구동하였다. 슬립모드와 액티브 모드에서 카스토퍼의 소모 전력은 각각 80mW와 25mW로 측정 되었다. 또한 구현된 시스템에서 항상 액티브 모드로 동작하는 XBee 모듈에 의해 발생하는 전력 소모 문제를 해결할 수 있는 초저전력 IoT 카스토퍼 구조를 제안하였다. 구현된 주차안내 시스템은 빅데이터 및 인공지능 기술과 연계되어 효율적인 주차관리 시스템의 개발에 활용될 수 있을 것으로 기대된다^[26,27].

References

- [1] <http://sfpark.org/>
- [2] <https://www.areaverda.cat/en>
- [3] <http://www.libelium.com/>
- [4] <http://www.nwave.io/sparkit-parking-sensor/>
- [5] <https://www.pnicorp.com/>
- [6] <https://www.smartparking.com/>
- [7] <http://www.parkingcloud.co.kr>
- [8] <https://www.ino-on.co.kr/>
- [9] <http://jndsystem.com/>
- [10] <https://www.magotec.com>
- [11] Q. Cao, T. Abdelzاهر, T. He, and J. Stankovic, "Towards optimal sleep scheduling in sensor networks for rare-event detection," in Proc. 4th Intl. Symp. Information Processing in Sensor Networks (IPSN), 2005.
- [12] <http://www.zigbee.org/>
- [13] Atmel, "8-bit Atmel with 8KBytes In-System Programmable Flash", ATmega8 datasheet.
- [14] <https://www.digi.com/1p/xbee>
- [15] Sharp, "General Purpose Type Distance Measuring Sensor," GP2Y0A21YK/GP2Y0D21YK datasheet.
- [16] Samsung SDI, "Specification of Product for Li-ion Rechargeable Cell," ICR18650 datasheet.
- [17] Texas Instruments, "800-mA Low-Dropout Linear Regulator," LMI117 datasheet.
- [18] <https://www.raspberrypi.org/>
- [19] <https://curl.haxx.se/libcurl/c/>
- [20] <https://aws.amazon.com/>
- [21] <https://httpd.apache.org/>
- [22] <http://www.php.net/>
- [23] <https://www.mysql.com/>
- [24] <https://developer.android.com/studio>
- [25] Texas Instruments, "Using Nanotimers to Reduce IoT System Power Consumption by an Order of Magnitude," TPL5010/5011 application note.
- [26] Sun-Jin Oh, "Design of a Smart Application using Big Data," The Journal of The Institute of Internet, Broadcasting and Communication, vol. 15, no. 6, pp. 17-24, Dec. 2015.

DOI: <https://doi.org/10.7236/JIIBC.2015.15.6.17>

[27] Won-tae Lee and JangMook Kang, "A study on Model of Personal Information Protection based on Artificial Intelligence Technology or Service," The Journal of The Institute of Internet, Broadcasting and Communication, vol. 16, no. 4, pp. 1-6, Aug. 2016.

DOI: <https://doi.org/10.7236/JIIBC.2016.16.4.1>

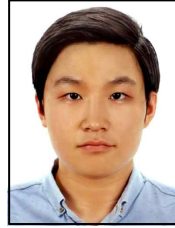
이 현 민(준회원)



- 2015년 : 서울과학기술대학교 화공생명공학과 (학사)
- 2017년 ~ 현재 : 서울과학기술대학교 스마트생산융합시스템공학과 (재학)

<주관심분야: 연료전지>

손 정 기(준회원)



- 2014년 : 서울과학기술대학교 MSDE 전공 (재학)

<주관심분야: 임베디드SW>

저자 소개

심 동 하(정회원)



- 1996년 : 서울대학교 원자핵공학과 (학사)
- 1998년 : 서울대학교 전기공학부 (석사)
- 1998년 ~ 2005년: 삼성전자 종합기술원
- 2011년 : University of Florida, ECE Dept. (박사)
- 2011년 ~ 현재: 서울과학기술대학교 MSDE전공 조교수

<주관심분야: 임베디드시스템>

양 지 훈(준회원)



- 2015년 : 가톨릭관동대학교 의료공학과 (학사)
- 2015년 ~ 현재 : 서울과학기술대학교 스마트생산융합시스템공학과 (재학)

<주관심분야: 임베디드시스템>

한 승 한(준회원)



- 2016년 : 서울과학기술대학교 전자IT 미디어공학과 (학사)
- 2017년 ~ 현재: 서울과학기술대학교 스마트생산융합시스템공학과 (재학)

<주관심분야: 임베디드시스템>

※ 이 연구는 서울과학기술대학교 교내연구비의 지원으로 수행되었습니다.